

SISTEMI OPERATIVI

ESERCIZIO N. 1 del 2 FEBBRAIO 2001

In una piccola località sciistica è in funzione una seggiovia, con una cabina che può contenere al massimo MAX sciatori. La seggiovia effettua due fermate: una a valle e una a monte. La cabina parte dalla fermata a valle solo quando ha caricato almeno MIN sciatori, mentre è sufficiente anche solo uno sciatore per partire dalla fermata a monte. Ogni sciatore, dopo essere entrato nella cabina a valle, aspetta che questa arrivi a monte per uscire; quando ha finito di sciare, riprende la seggiovia e attende che la cabina arrivi a valle per uscire. Si implementi una soluzione usando il costrutto monitor per modellare la seggiovia e i processi per modellare gli sciatori e la cabina e si descriva la sincronizzazione tra i processi. Nella soluzione si massimizzi l'utilizzo delle risorse. Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare starvation.

```
program impianto_sciistico

const MIN = ...; { numero minimo di sciatori }
const MAX = ...; { capacità della cabina }
type dir = (su, giù, viaggio); { stato della cabina }

type sciatore = process
begin
    repeat
        seggiovia.entra(su);
        < preparati ad uscire >
        seggiovia.esci(su);
        < scia >
        seggiovia.entra(giù);
        < preparati ad uscire >
        seggiovia.esci(giù);
    until false
end

type cabina = process
begin
    repeat
        seggiovia.carica(su);
        < fai il viaggio in salita >
        seggiovia.scarica(su);
        seggiovia.carica(giù);
        < fai il viaggio in discesa >
        seggiovia.scarica(giù);
    until false
end

type seggiovia_m = monitor

{ variabili del monitor }
var stato : dir;
    { stato della cabina }
    n_sciatori : integer;
```

```

    { numero di sciatori in cabina }
    coda : array[2] of condition;
    { code su cui sospendere in attesa della cabina }
    coda_viaggio : condition;
    { sciatori in cabina in attesa di arrivare }
    coda_cabina : condition;
    { cabina in attesa di partire }

procedure entry entra (d: dir)
begin
    if stato <> d and n_sciatori < MAX then { se la cabina non è presente }
        coda[d].wait;
    n_sciatori ++ ; { sono entrato }
    { se abbiamo raggiunto il numero minimo }
    if (d = su and n_sciatori >= MIN) or (d = giù) then
        { faccio partire la cabina }
        coda_cabina.signal;
        { mi sospendo in attesa di arrivare }
    coda_viaggio.wait;
end

procedure entry esci (d: dir)
begin
    n_sciatori -- ; { siamo arrivati }
end

procedure entry carica (d: dir)
begin
    stato:=dir;
    { sveglio gli sciatori in attesa di entrare }
    for i := 1 to MAX do
        coda[d].signal;
        { se non c'è il numero minimo di sciatori }
        if (d = su and n_sciatori < MIN) or
            (d = giù and n_sciatori < 1) then
            coda_cabina.wait;
        stato := viaggio; { ripartiamo! }
    end
end

procedure entry scarica (d: dir)
begin
    { sveglio tutti gli sciatori in attesa di uscire }
    s:=n_sciatori;
    for i := 1 to s do
        coda_viaggio.signal;
    end
end

function other (d: dir) : dir
begin
    if d = su then other := giù; else other := su;
end

begin { inizializzazione delle variabili }
    n_sciatori := 0;
    stato := giù;
end

var seggiovia: seggiovia_m; { il nostro monitor }
s1, s2, ... : sciatore;

```

```
c: cabina;
```

```
begin end.
```

Starvation

La soluzione proposta presenta starvation nel caso in cui gli sciatori a valle debbano aspettare che la cabina sia utilizzata dagli sciatori a monte, e viceversa. È un tipo di starvation legato al problema stesso. Si può risolvere imponendo un tempo massimo di attesa per la cabina, dopodiché effettua il viaggio anche senza sciatori.